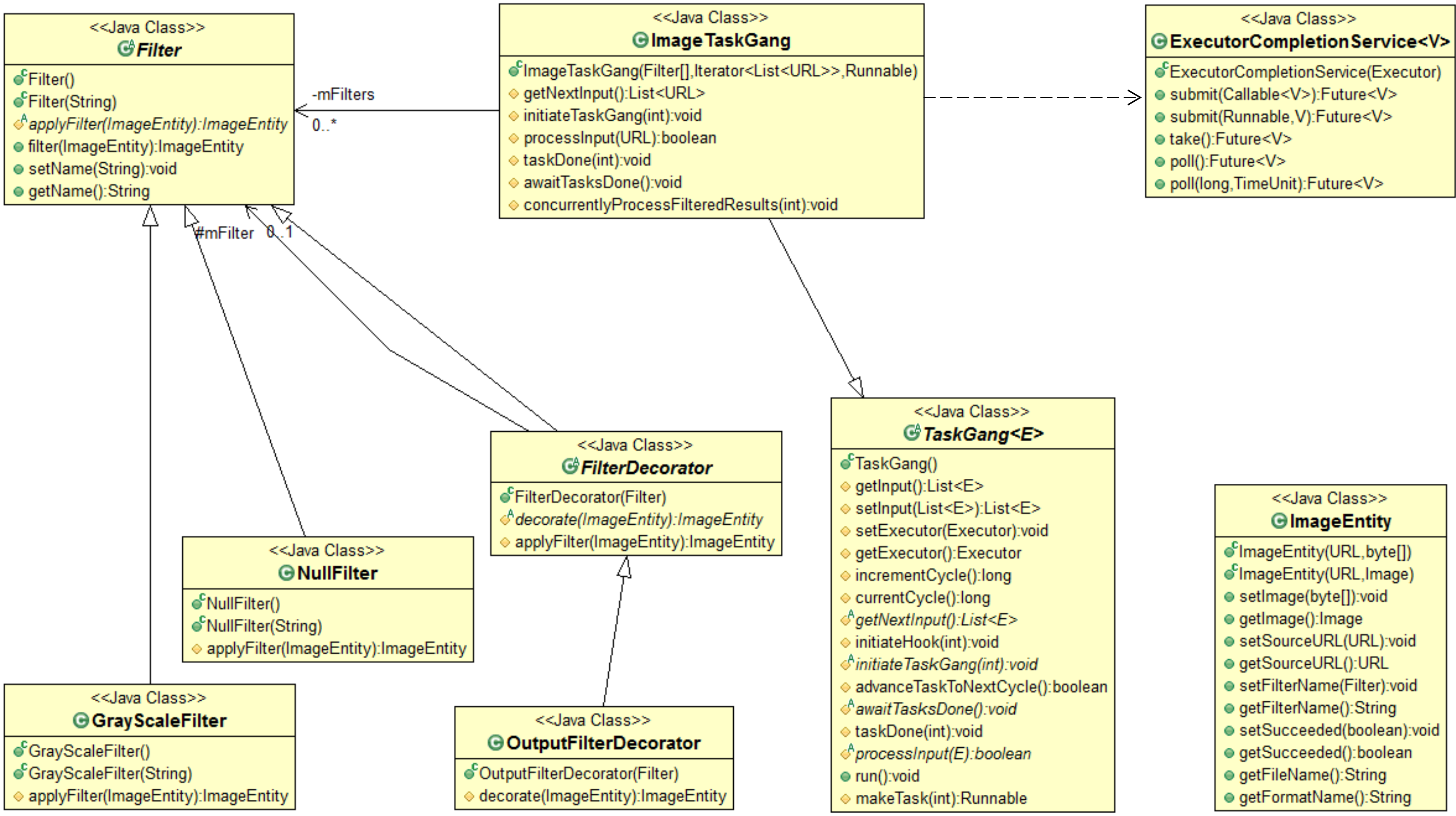
Structure & Dynamics of the ImageTaskGang Application

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the structure & dynamics of the ImageTaskGang applications

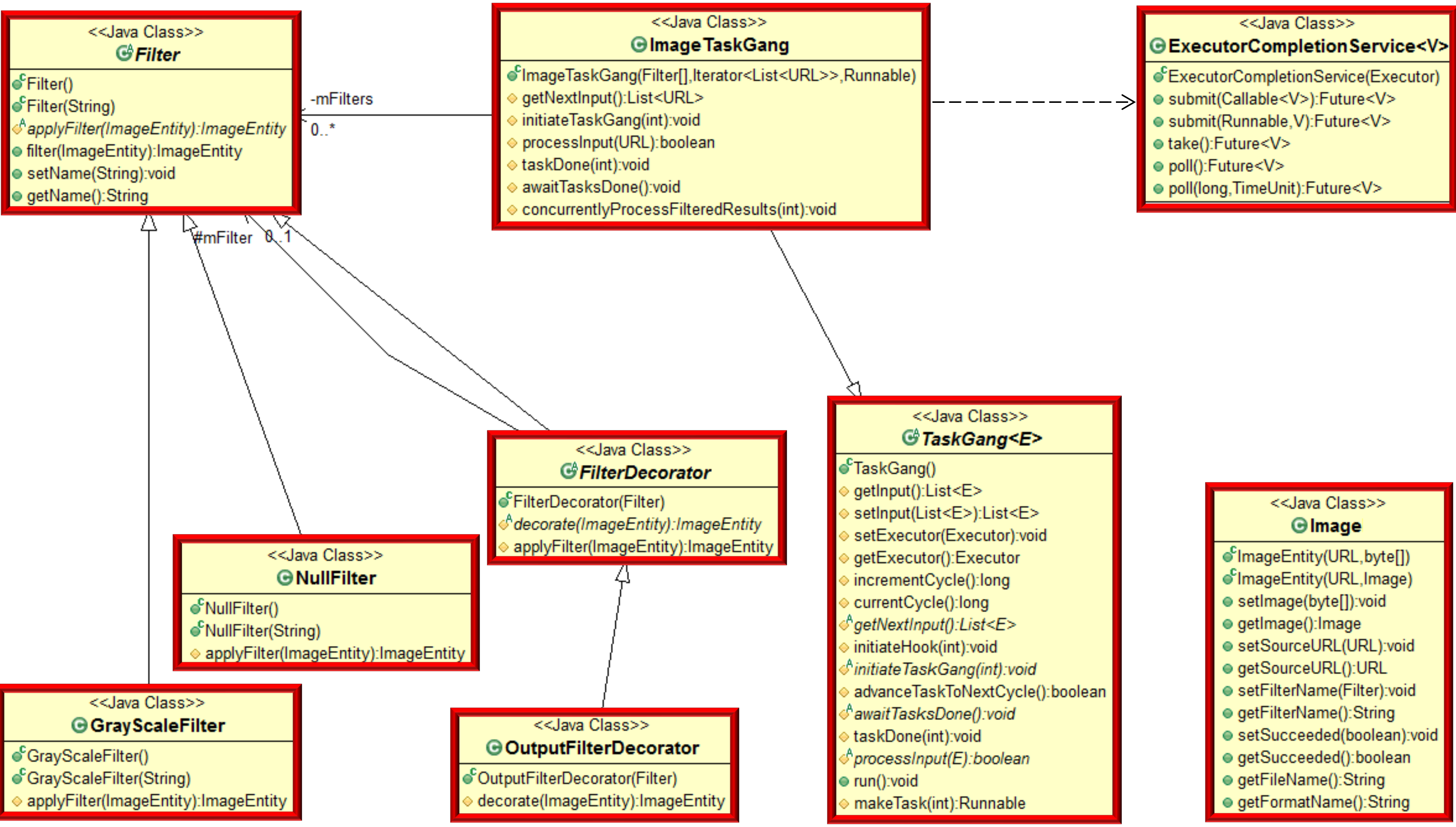


See github.com/douglascraigschmidt/LiveLessons/tree/master/ImageTaskGang

The Structure of the ImageTaskGang Application

The Structure of the ImageTaskGang Application

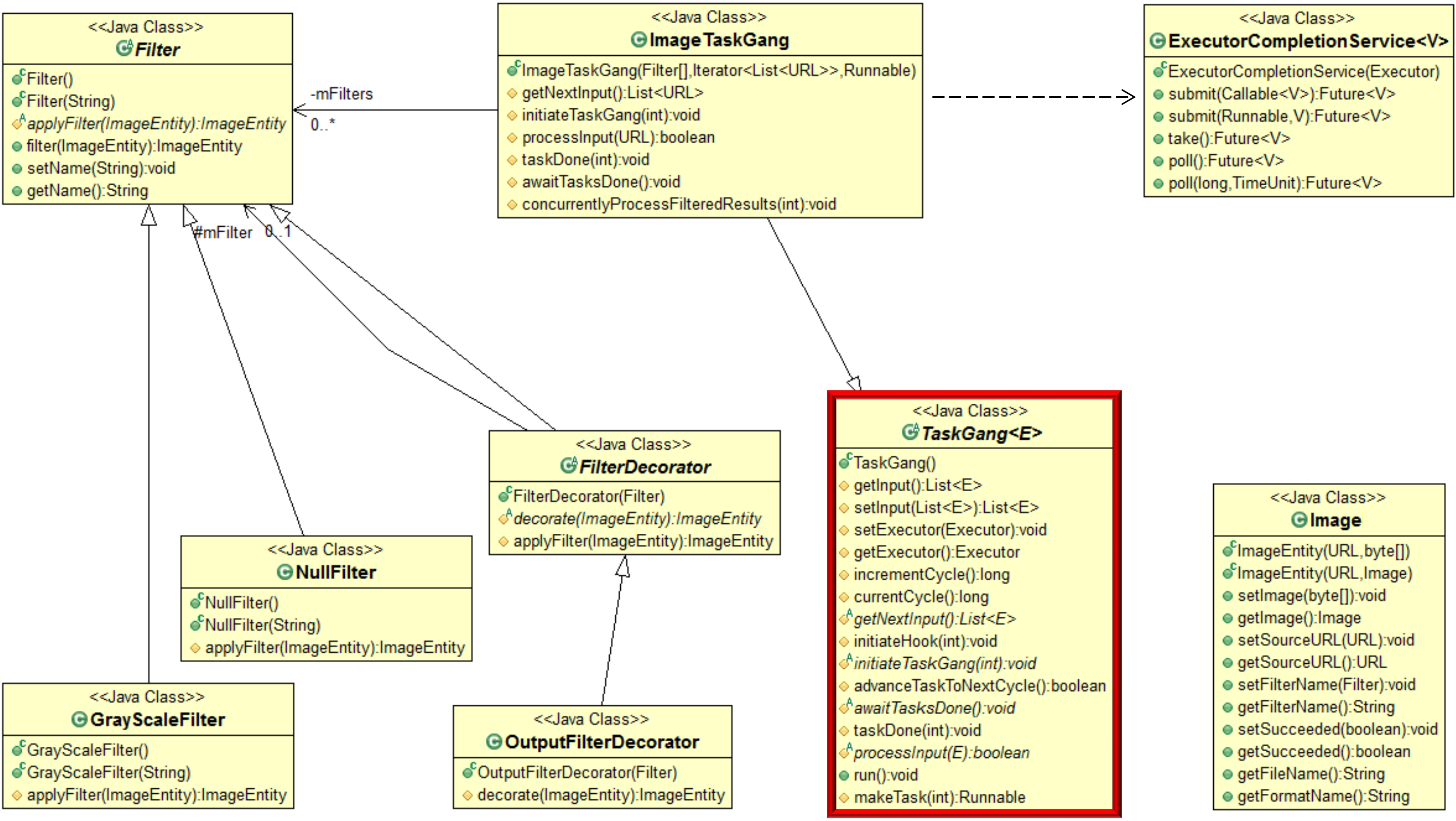
- UML class diagram for key components in the ImageTaskGang application



These classes implement the application's concurrency engine

The Structure of the ImageTaskGang Application

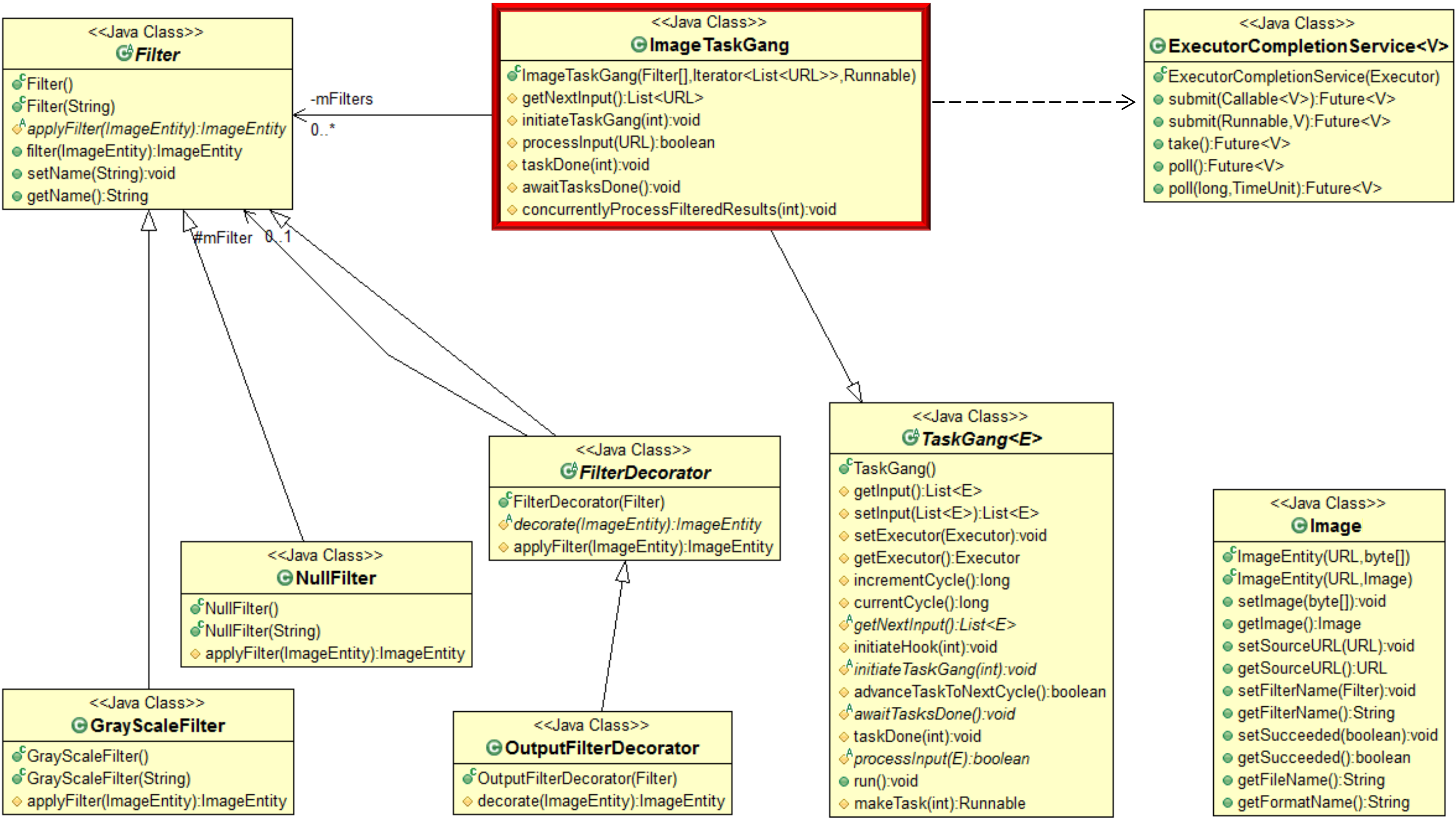
- UML class diagram for key components in the ImageTaskGang application



Defines a framework for spawning & running a "gang" of tasks that concurrently process input from a generic list

The Structure of the ImageTaskGang Application

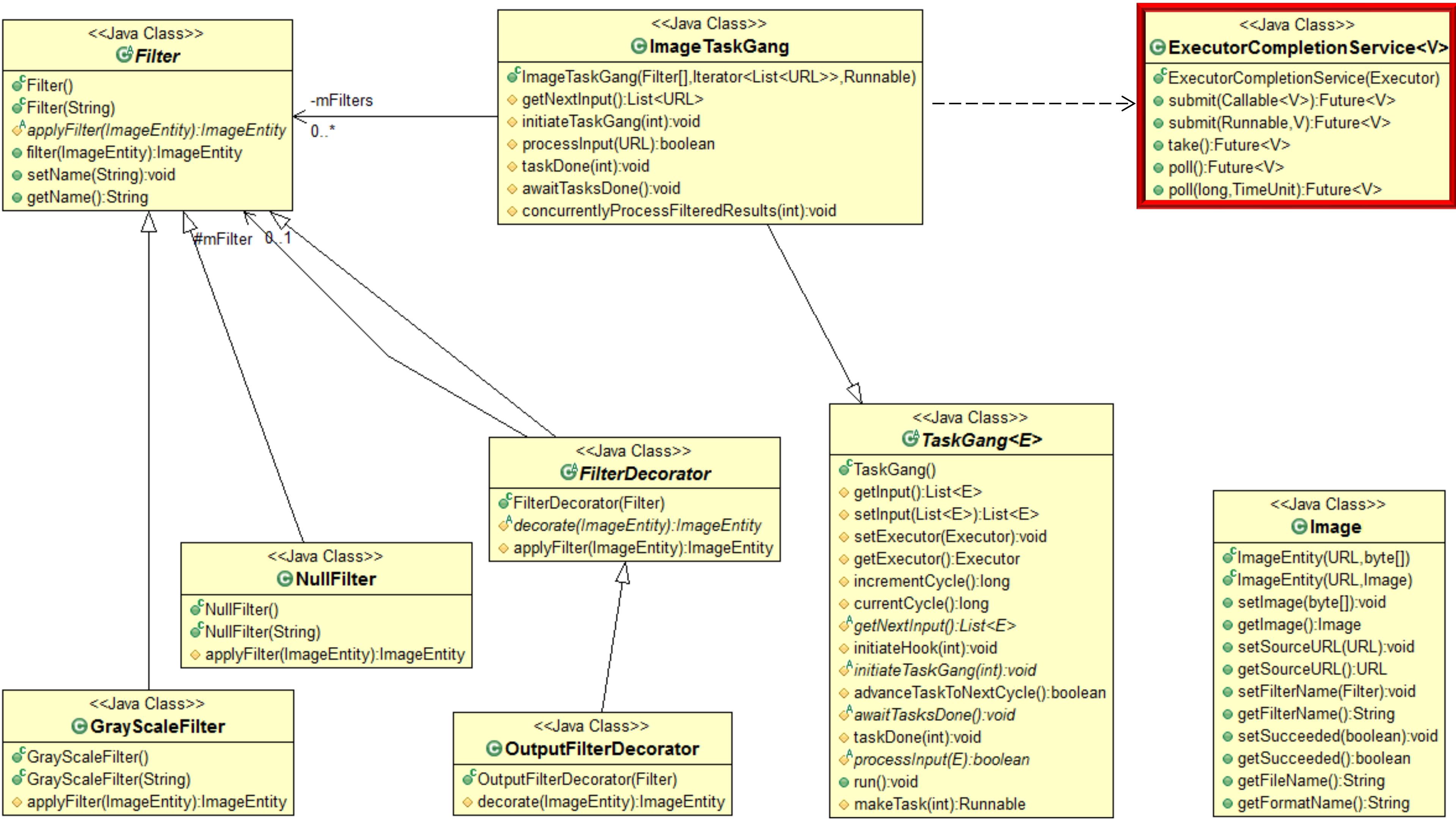
- UML class diagram for key components in the ImageTaskGang application



This class customizes the TaskGang framework for image processing

The Structure of the ImageTaskGang Application

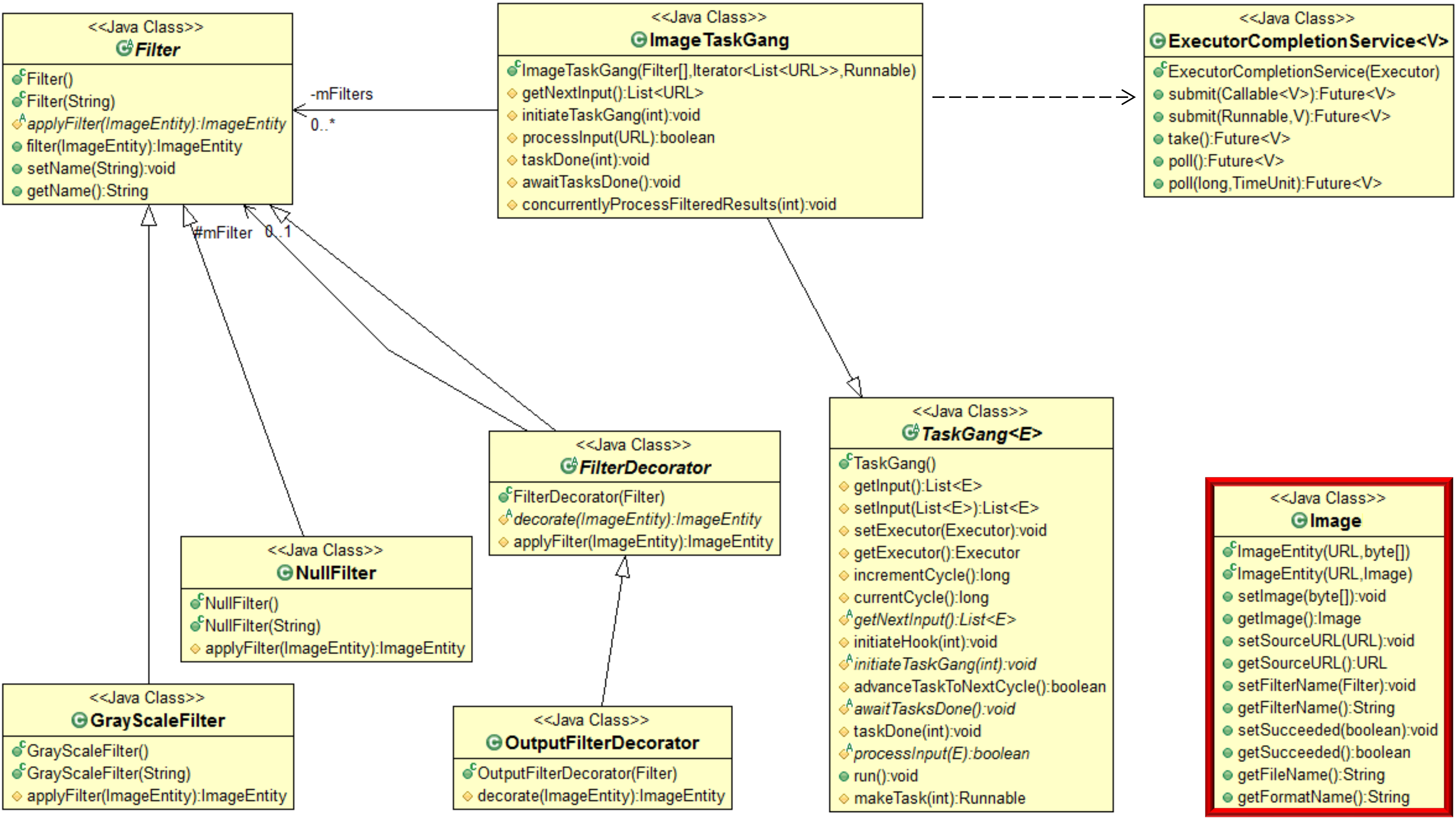
- UML class diagram for key components in the ImageTaskGang application



This concurrent Java class can be used to implement the *Proactor* pattern

The Structure of the ImageTaskGang Application

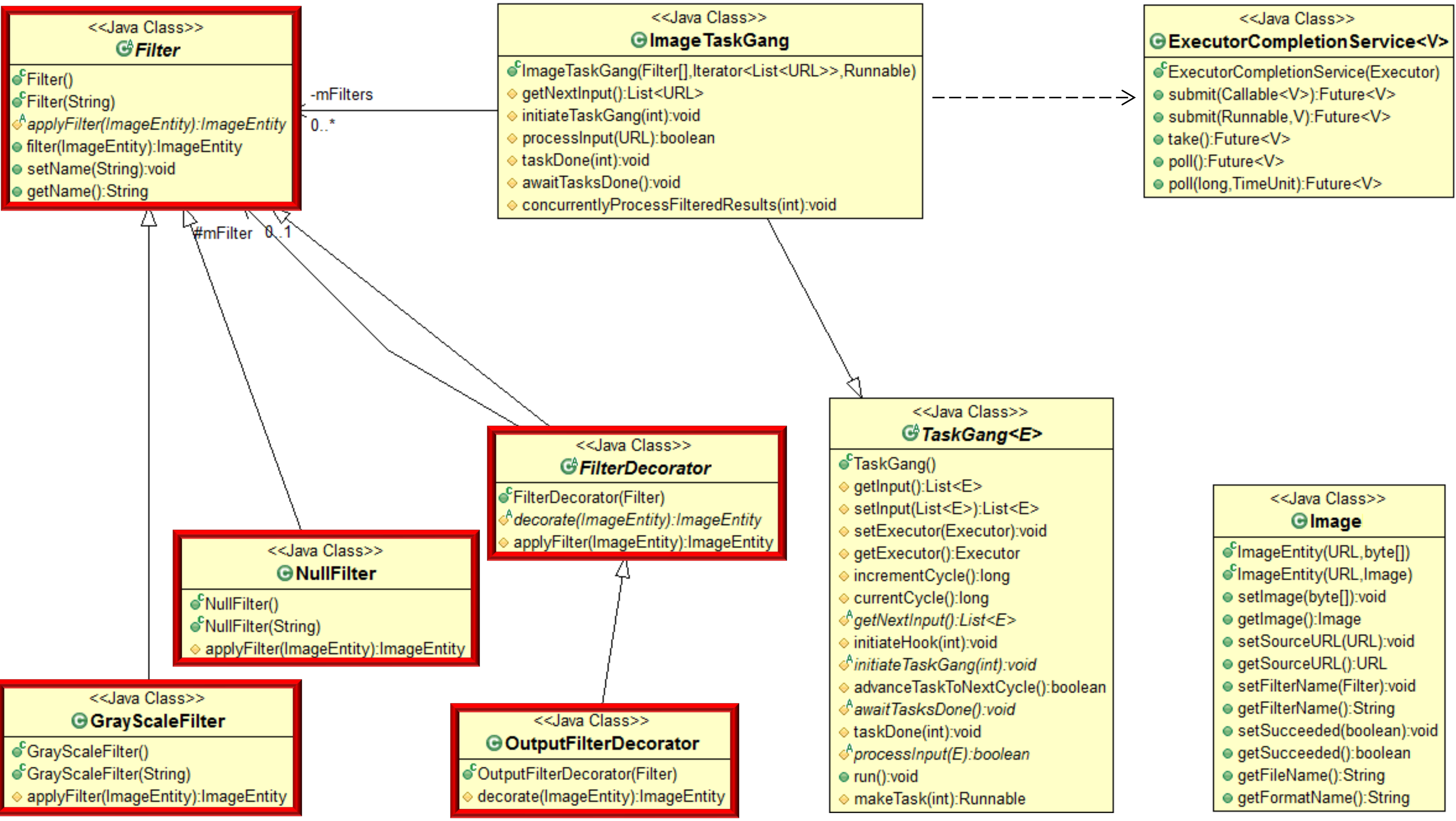
- UML class diagram for key components in the ImageTaskGang application



This class stores meta-data about an image & enables image processing

The Structure of the ImageTaskGang Application

- UML class diagram for key components in the ImageTaskGang application

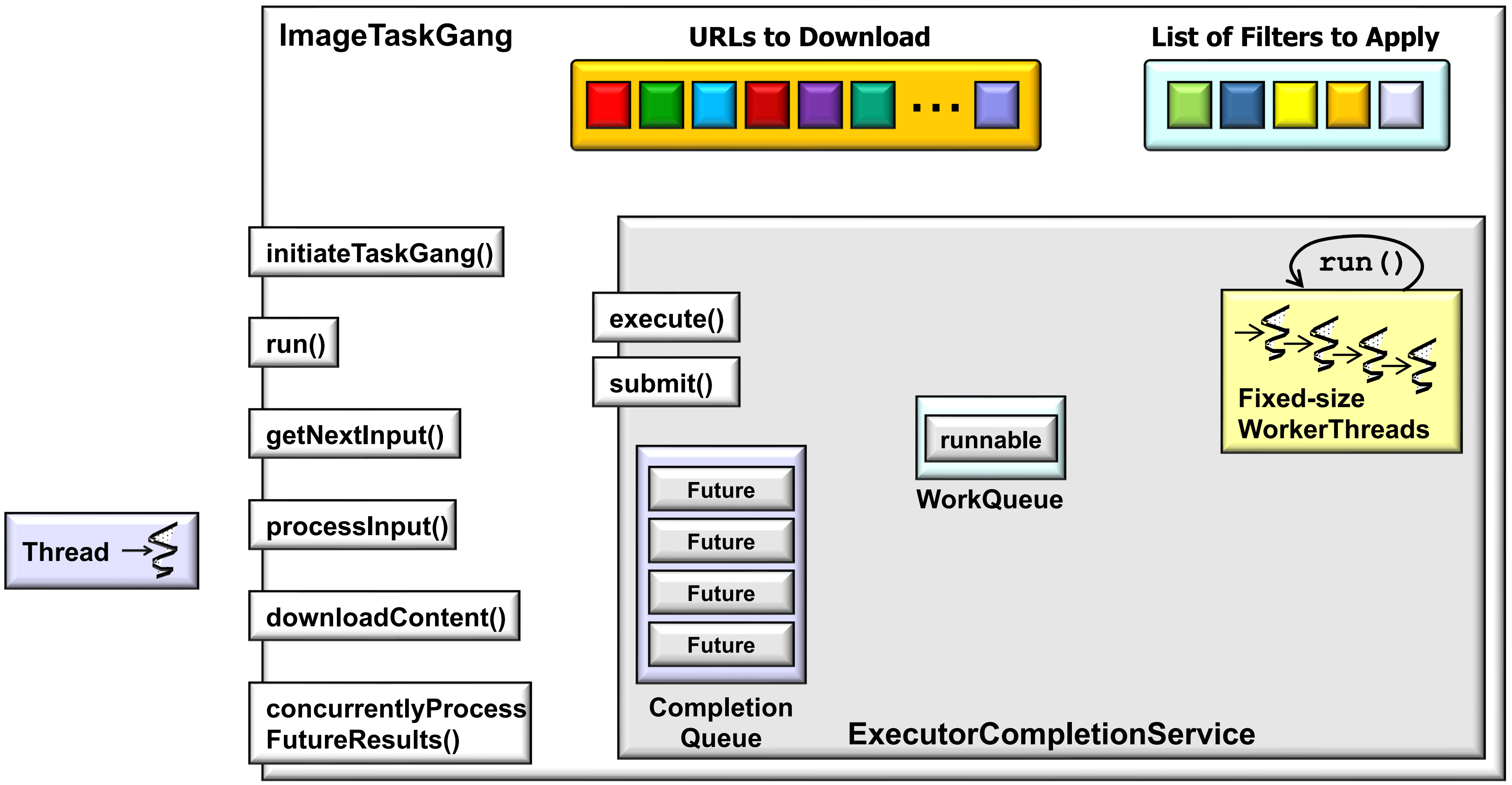


These classes implement image filters via the *Decorator* pattern

The Dynamics of the ImageTaskGang Application

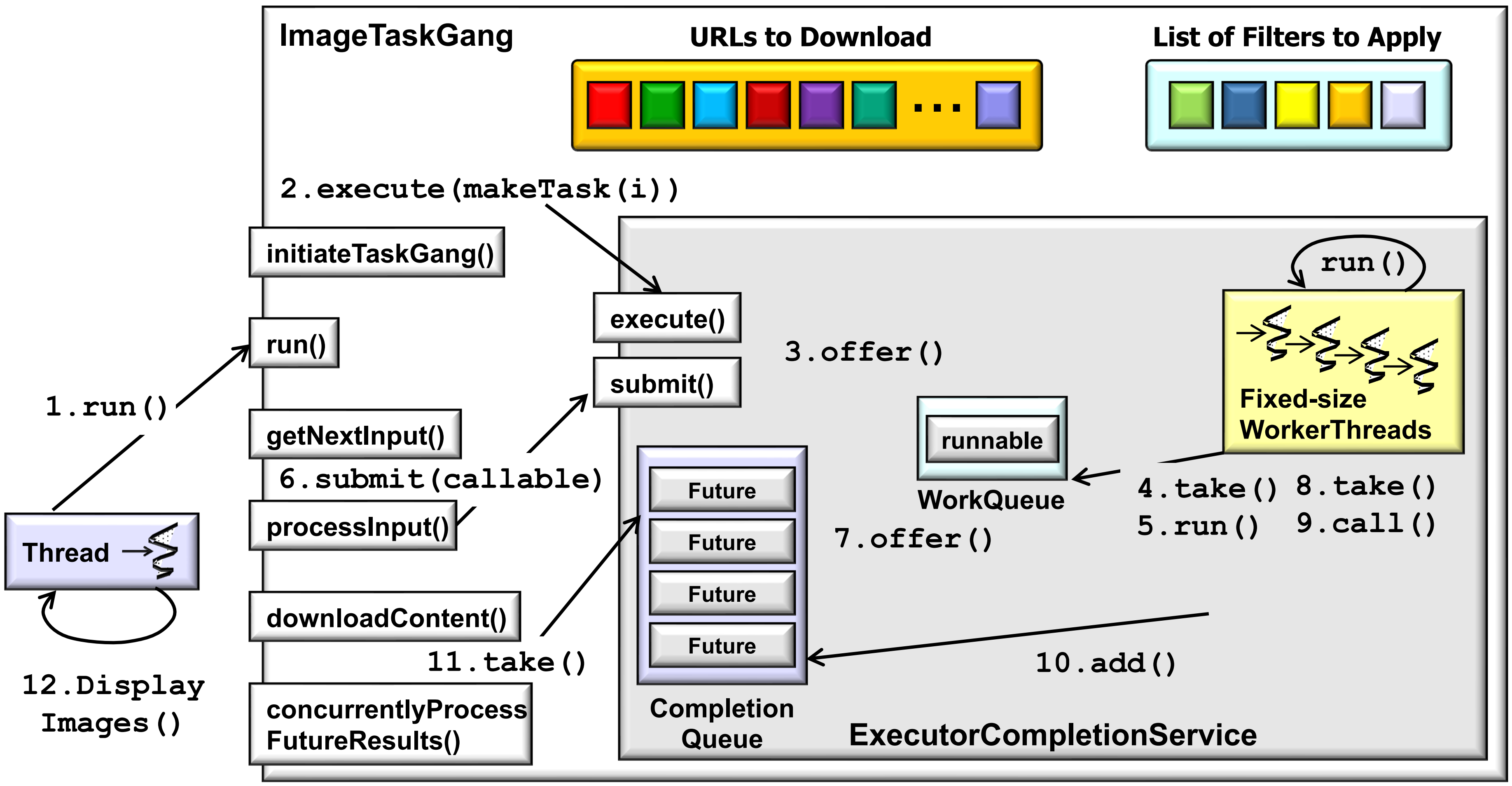
The Dynamics of the ImageTaskGang Application

- Object interaction diagram for the ImageTaskGang application



The Dynamics of the ImageTaskGang Application

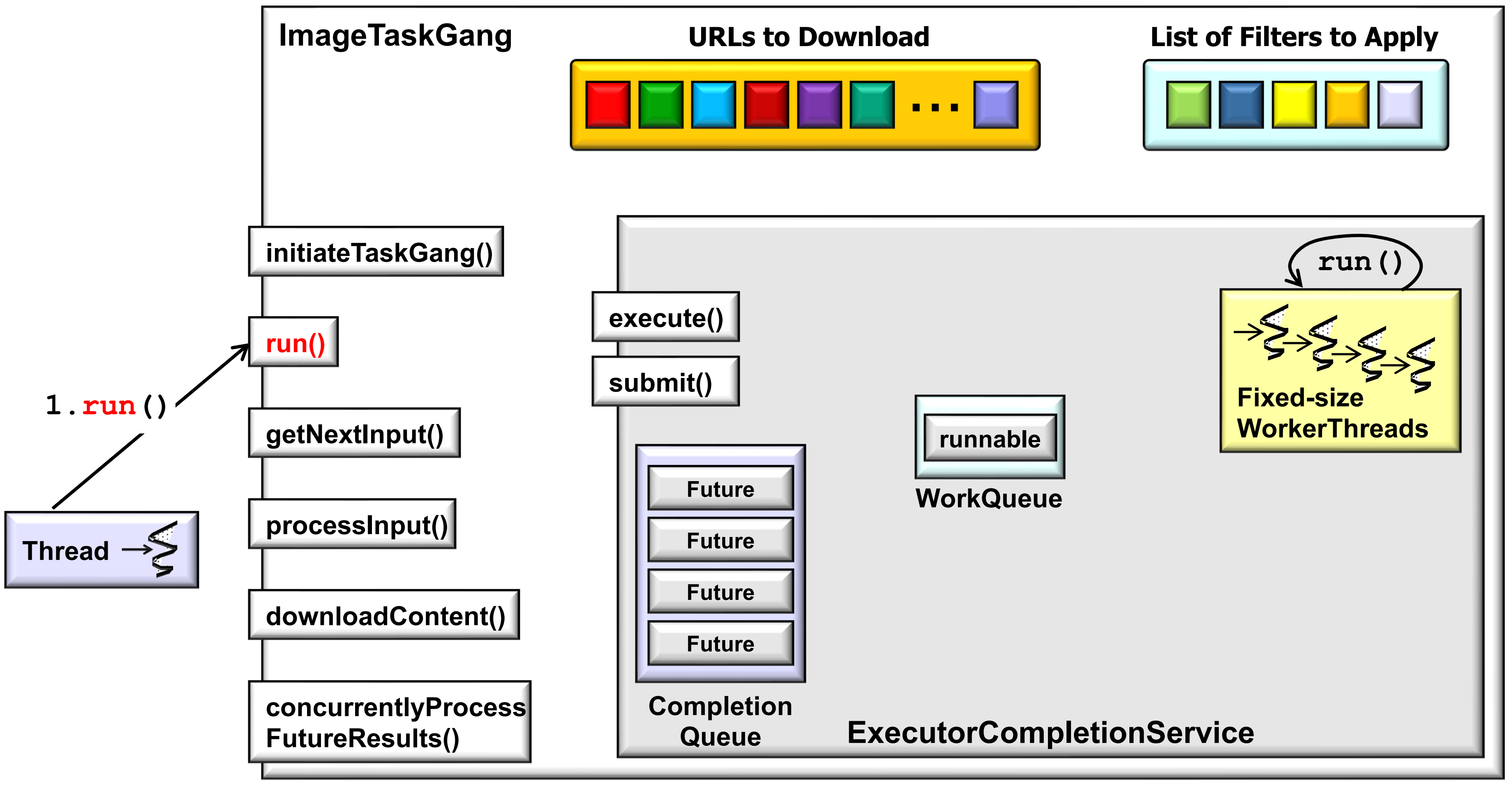
- Object interaction diagram for the ImageTaskGang application



Shows the steps used by the ImageTaskGang application to download, process, store, & display images from web servers

The Dynamics of the ImageTaskGang Application

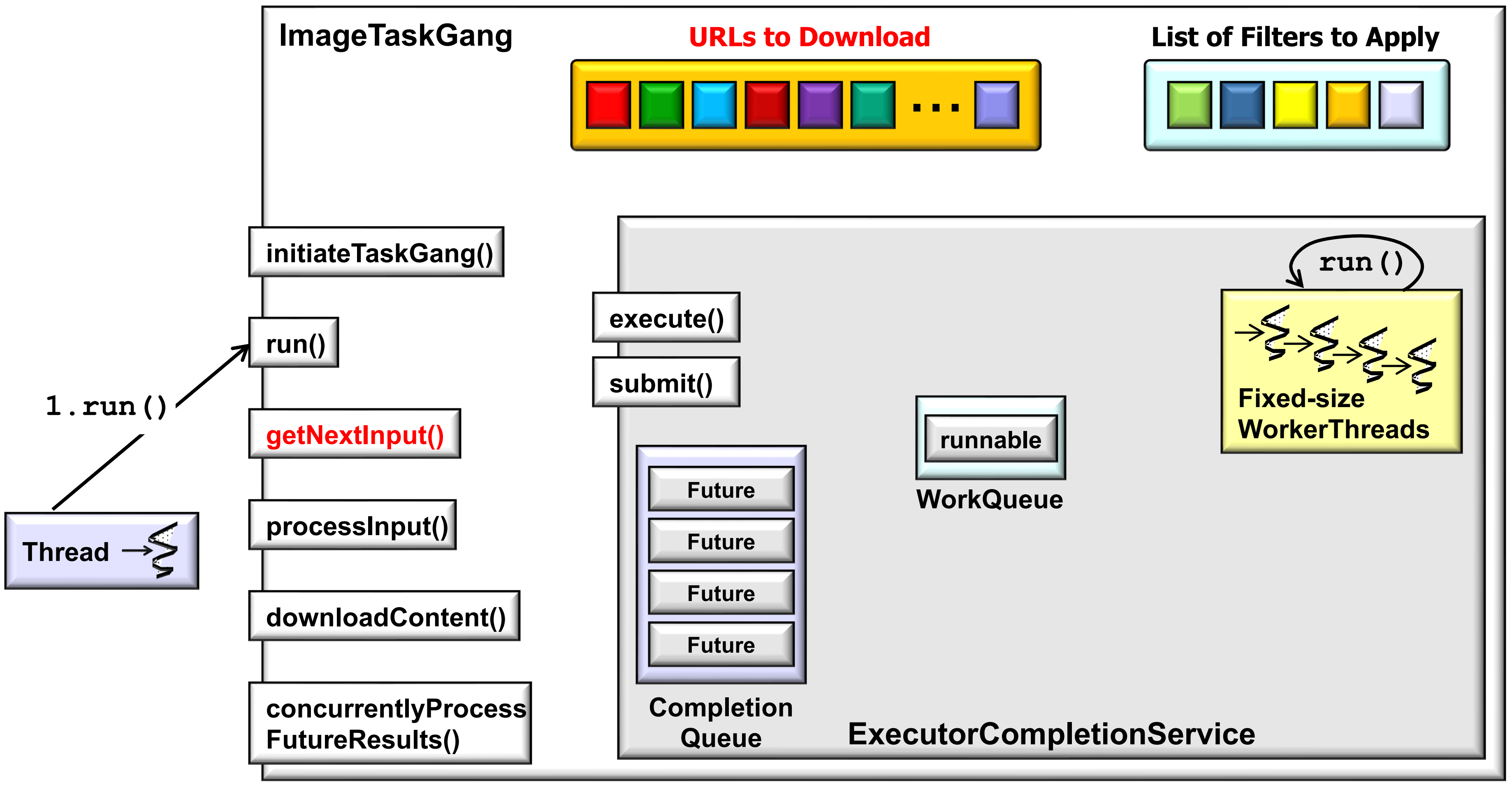
- Object interaction diagram for the ImageTaskGang application



The ImageTaskGang run() hook method obtains the list of URLs & creates Runnables to process them concurrently via Java's ExecutorCompletionService

The Dynamics of the ImageTaskGang Application

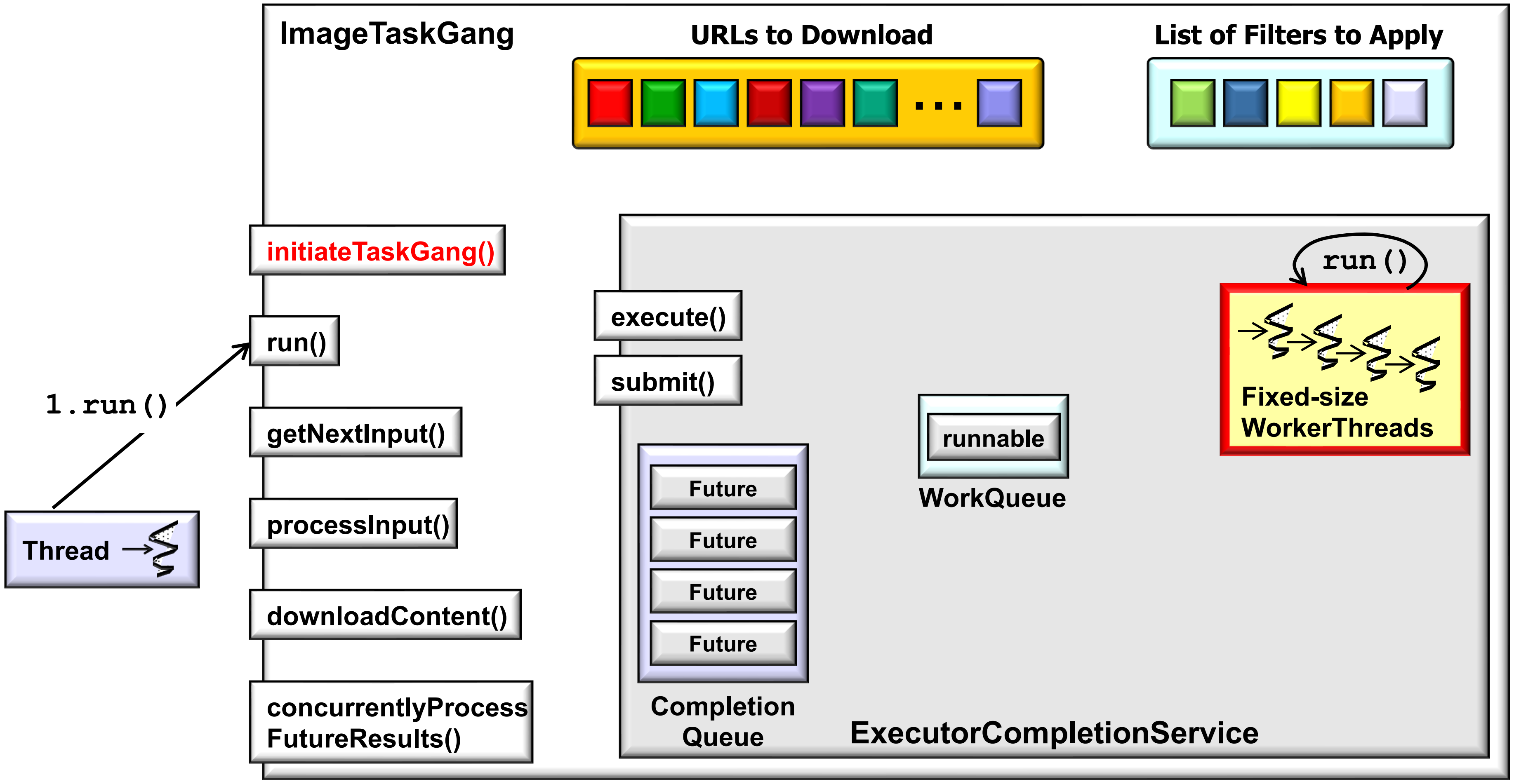
- Object interaction diagram for the ImageTaskGang application



getNextInput() retrieves the next tranche of URLs to download concurrently

The Dynamics of the ImageTaskGang Application

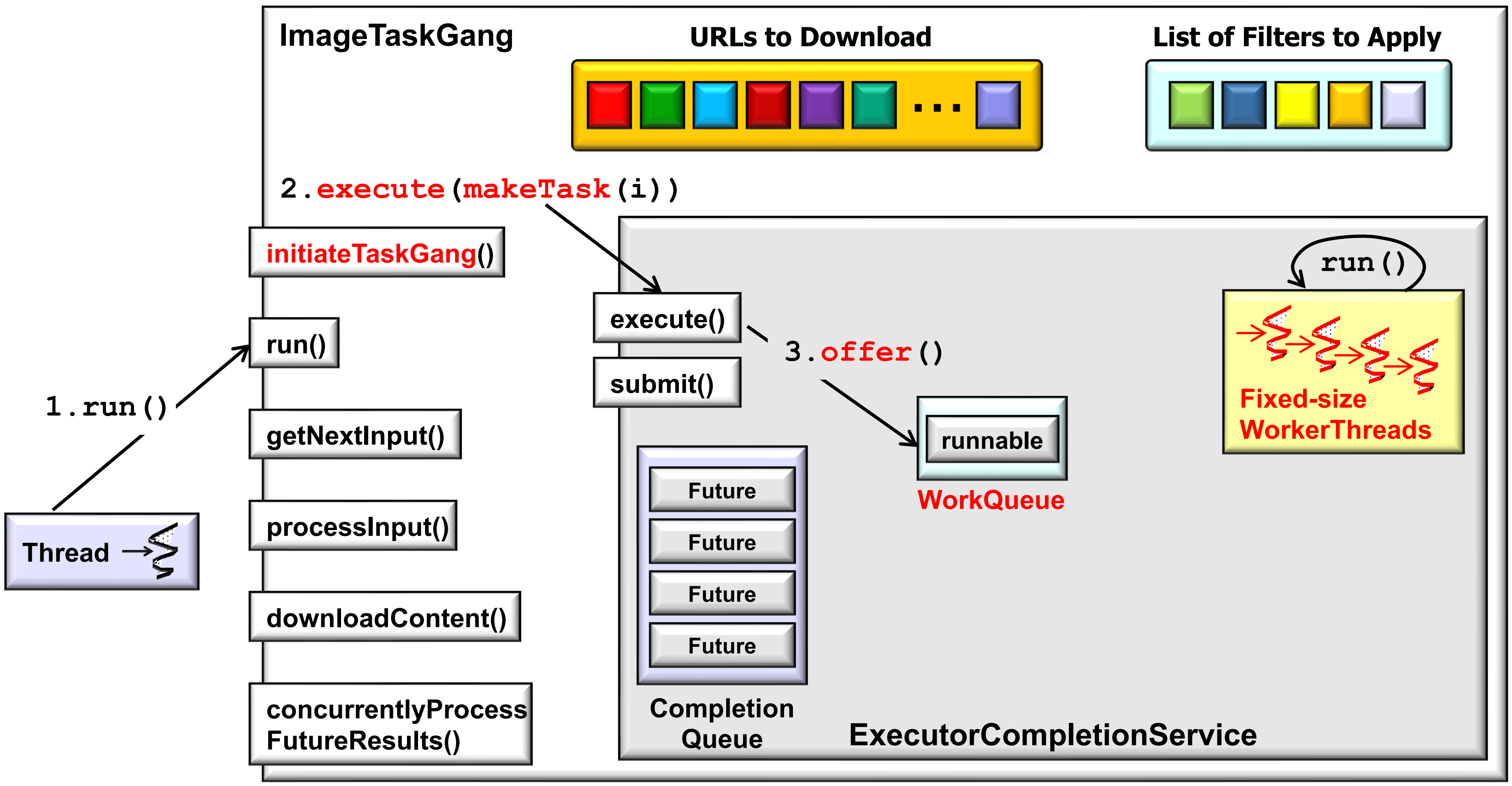
- Object interaction diagram for the ImageTaskGang application



initiateTaskGang() creates the designated type of thread pool

The Dynamics of the ImageTaskGang Application

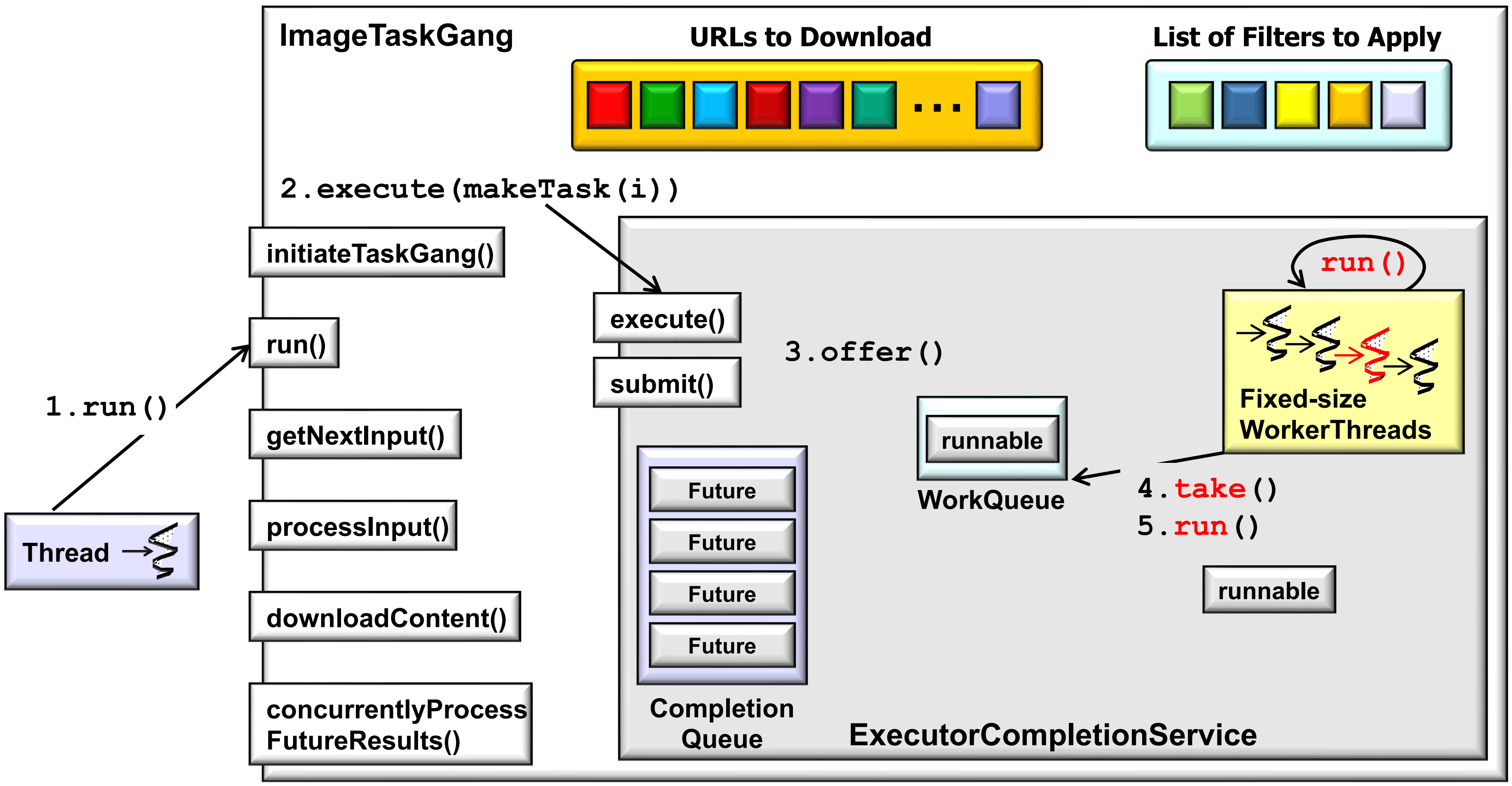
- Object interaction diagram for the ImageTaskGang application



A Runnable task is the created & scheduled to run via a fixed-size thread pool's work queue

The Dynamics of the ImageTaskGang Application

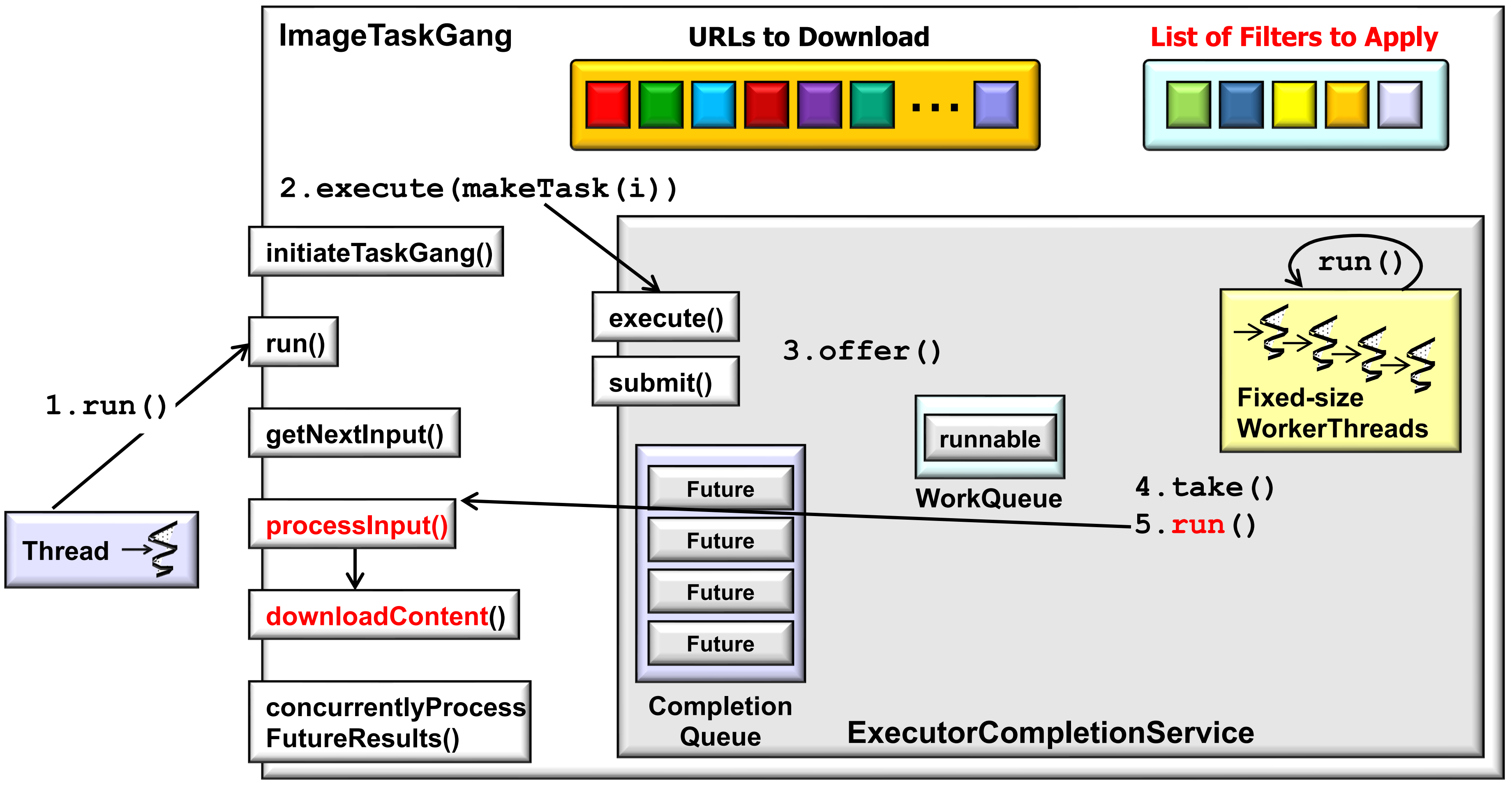
- Object interaction diagram for the ImageTaskGang application



One thread in that fixed-sized thread pool deques the task & runs it

The Dynamics of the ImageTaskGang Application

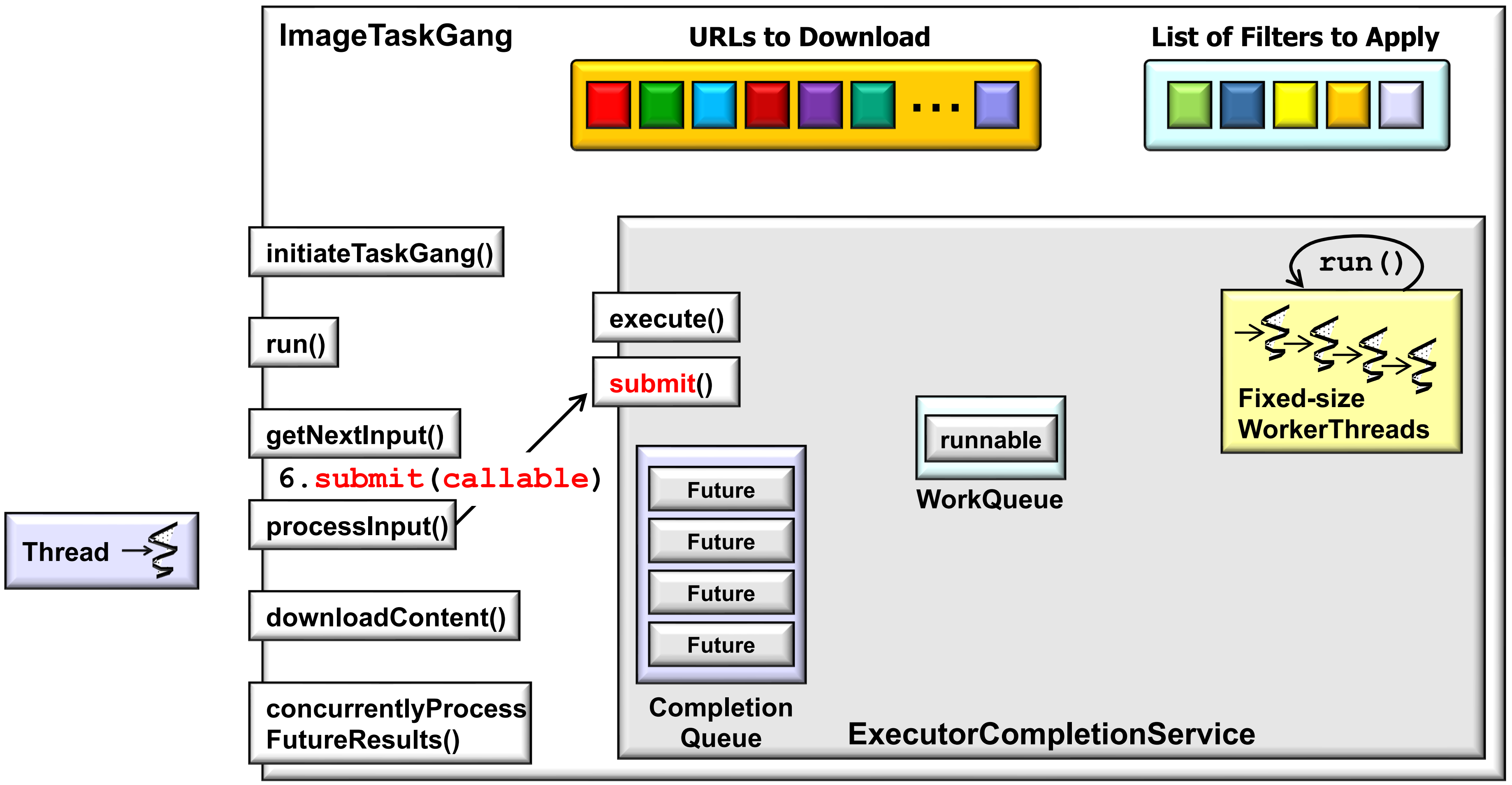
- Object interaction diagram for the ImageTaskGang application



A worker thread's run() hook method calls ImageTaskGang processInput(), which concurrently downloads each image & applies a list of filters to it

The Dynamics of the ImageTaskGang Application

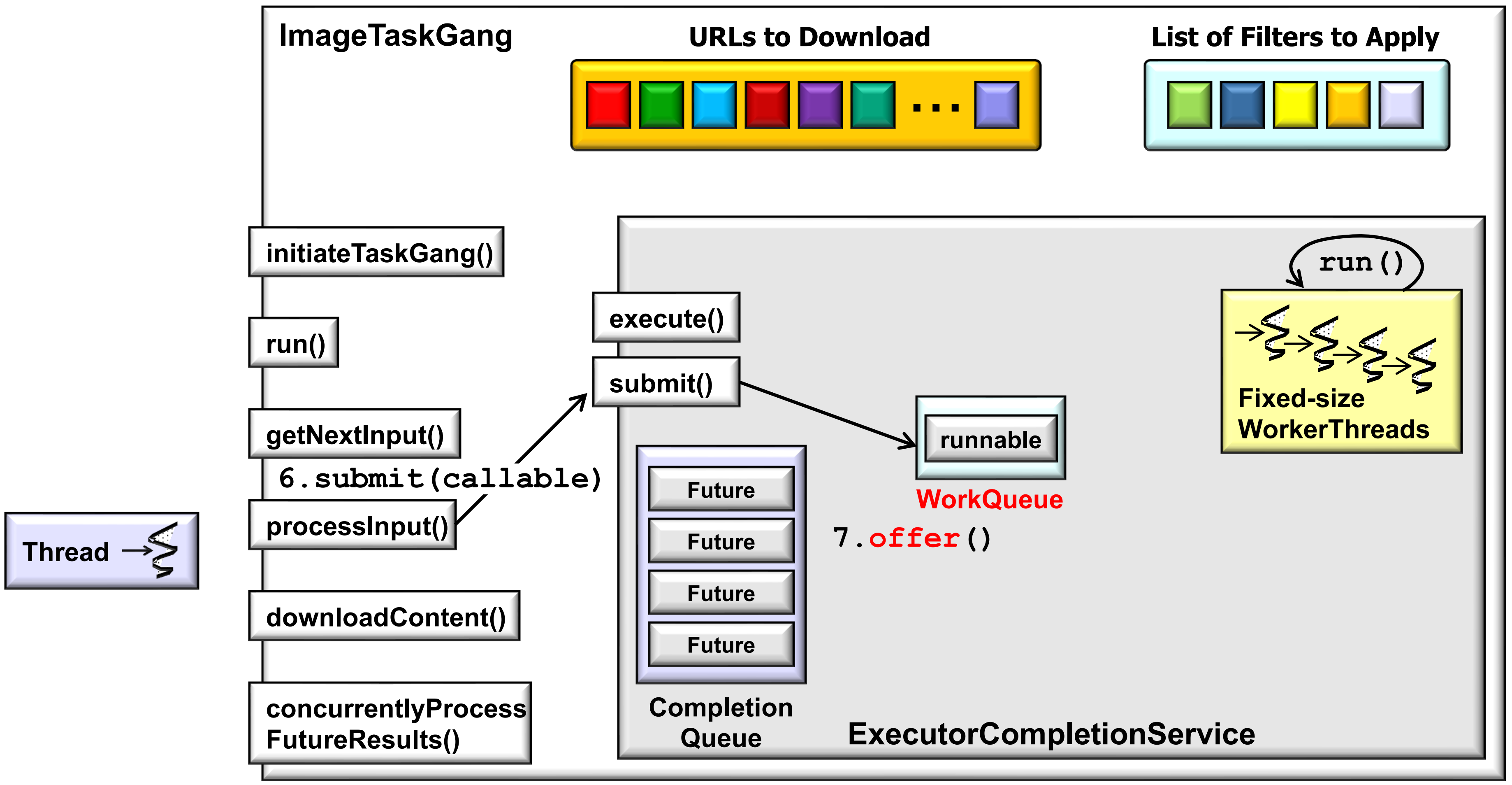
- Object interaction diagram for the ImageTaskGang application



Each filter operation is submitted to the Java ExecutorCompletionService as a callable lambda expression that will run asynchronously in the thread pool

The Dynamics of the ImageTaskGang Application

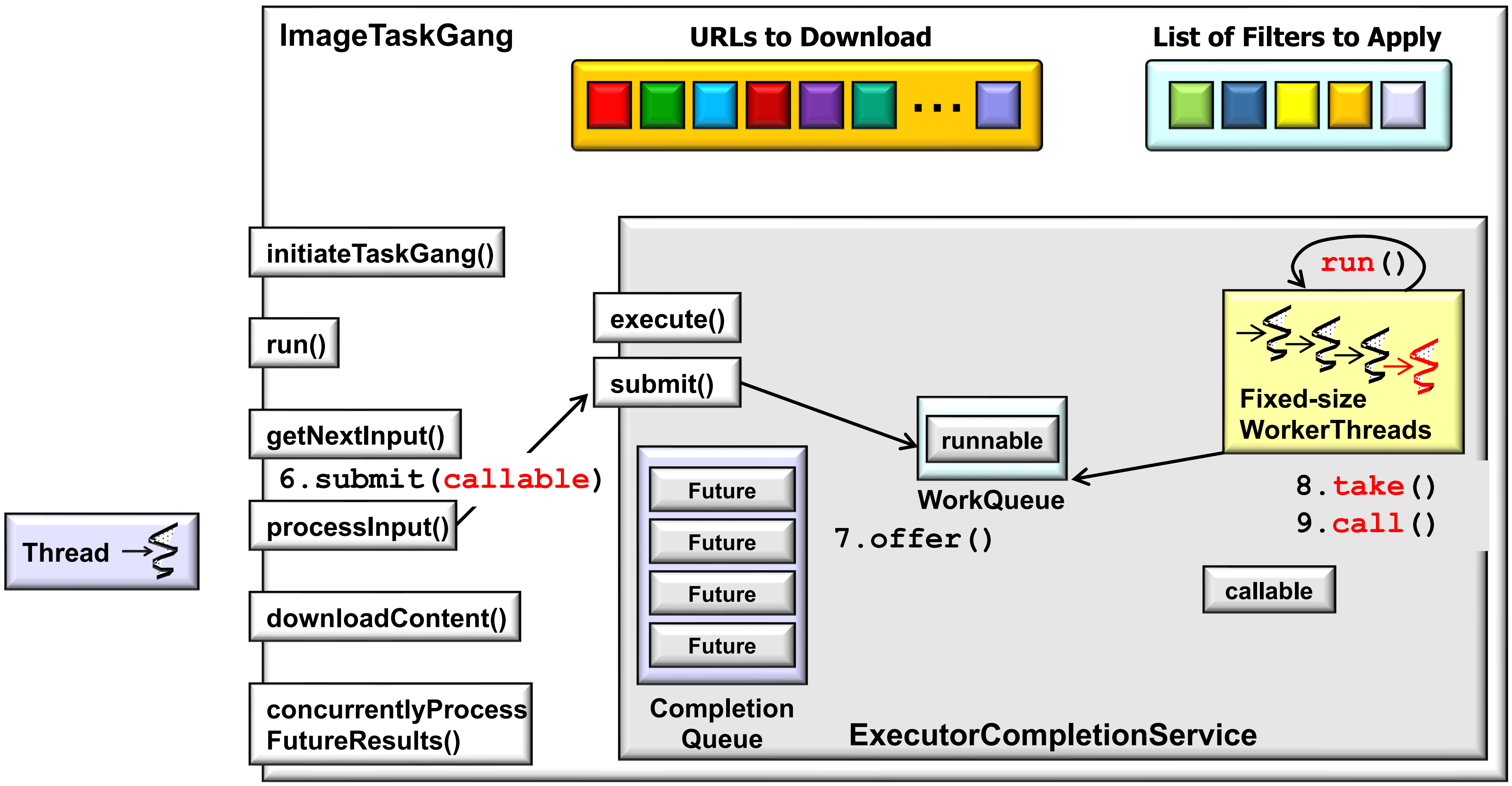
- Object interaction diagram for the ImageTaskGang application



A filter operation is also scheduled to run via the fixed-sized thread pool's work queue

The Dynamics of the ImageTaskGang Application

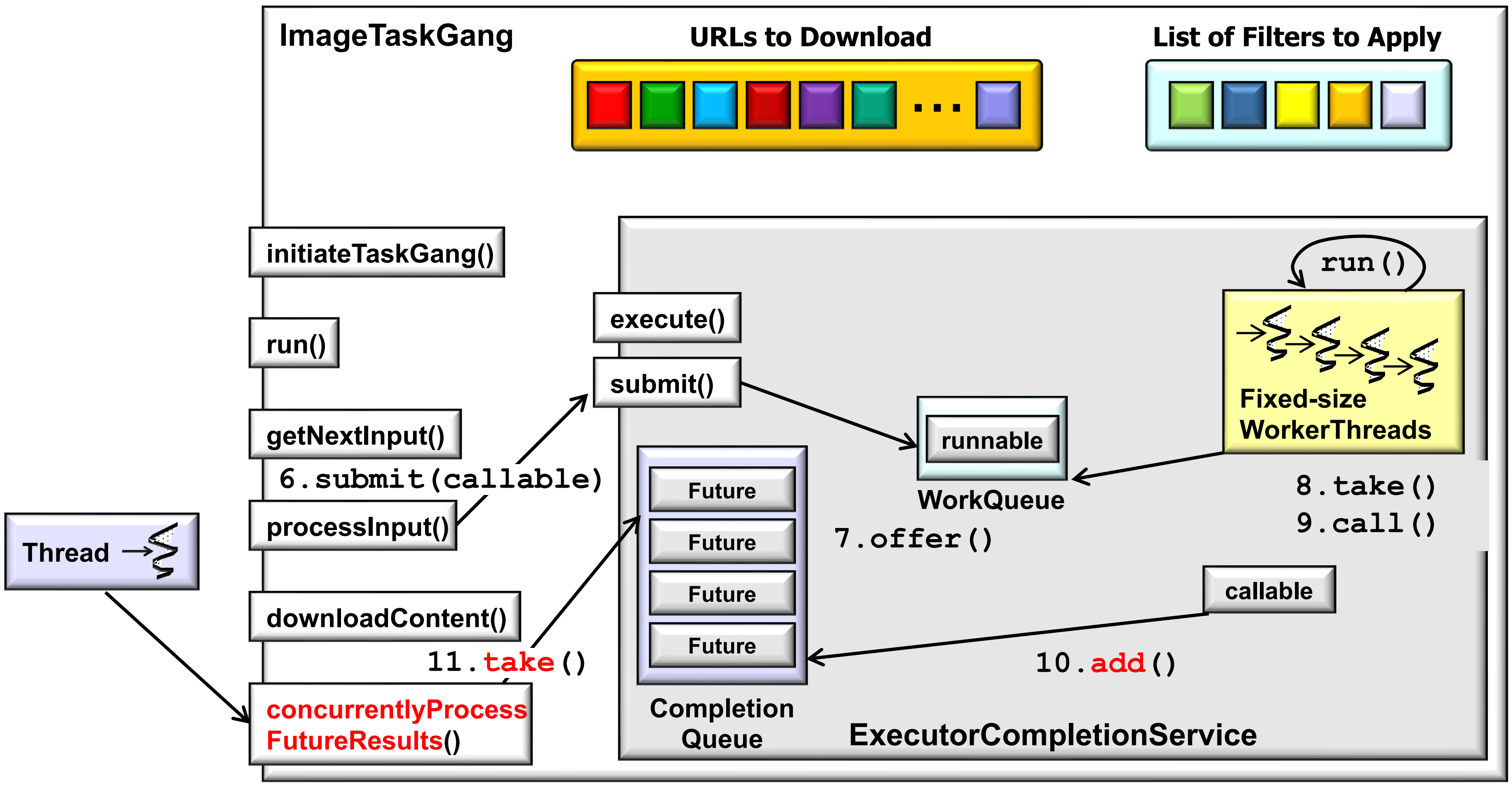
- Object interaction diagram for the ImageTaskGang application



A worker thread's run() hook method invokes the lambda's call() hook method, which filters the downloaded image & stores it in a local file

The Dynamics of the ImageTaskGang Application

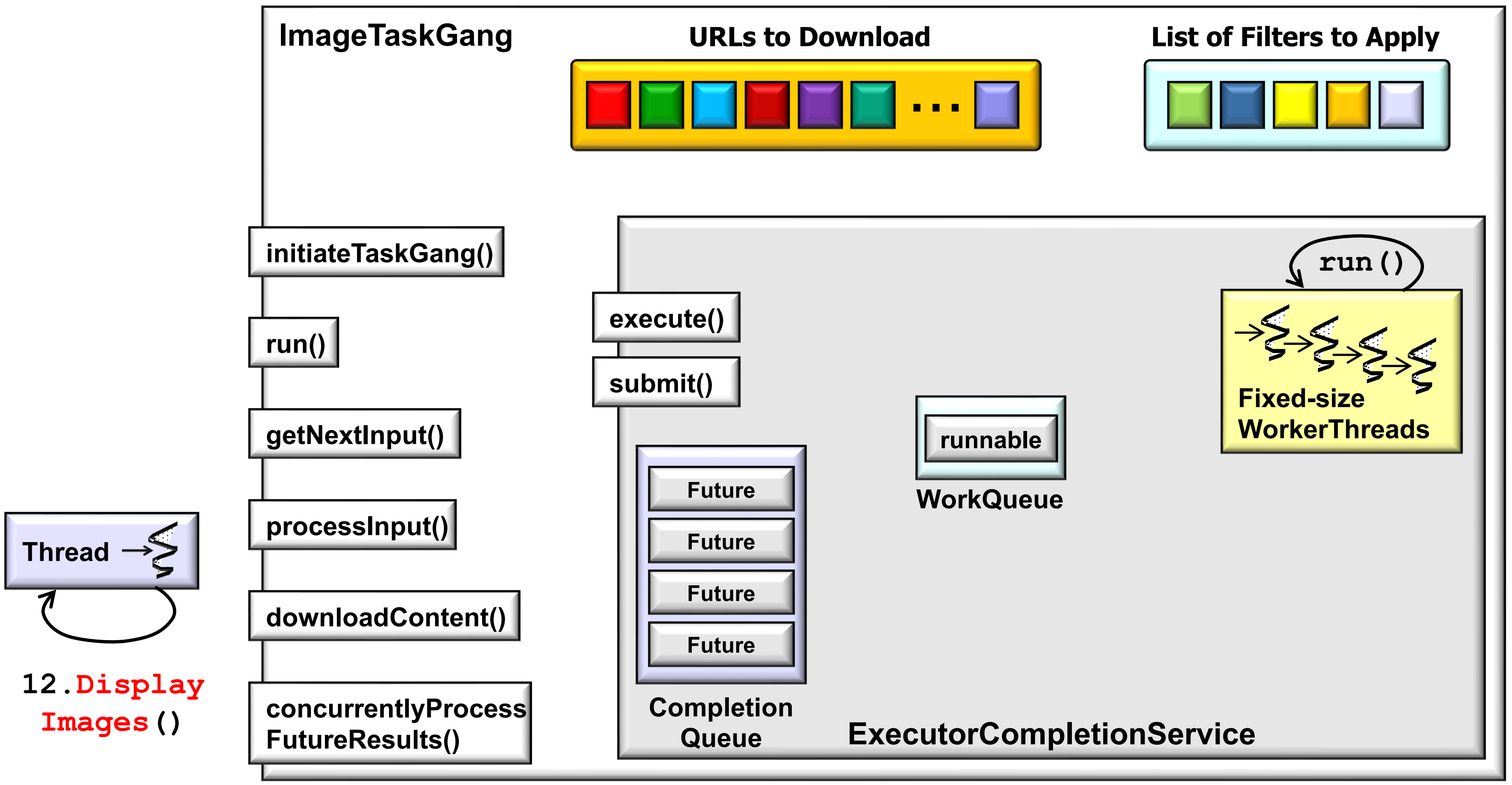
- Object interaction diagram for the ImageTaskGang application



The results of completed callable lambdas are queued & processed by the main thread

The Dynamics of the ImageTaskGang Application

- Object interaction diagram for the ImageTaskGang application



The main thread also triggers the displaying of images to the user after they are processed asynchronously

End of Structure & Dynamics of the Image TaskGang Application